We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS BOOK CITATION INDEX INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

1

A Bio-inspired Nano-Agent Architecture for Intelligent Agents

Jean-Claude Heudin
Interactive Media Lab. – IIM Léonard de Vinci
France

1. Introduction

Intelligent artificial creatures cover a large range of applications in various domains. Recent advances in intelligent agent technologies make now possible to develop a growing number of real-world applications. However, these applications require a new generation of open software architectures that combines such technologies with lightweight design and portability. This chapter describes a new nano-agent architecture designed for intelligent artificial creatures. This software environment takes advantages of our past experiences in distributed artificial intelligence with the Knowledge-based Operating System (Heudin et al., 1986), real-time multi-expert applications such as the Electronic Copilot project for combat aircrafts (Gilles et al., 1991), and the more recent Evolutionary Virtual Agent (EVA) applications (Heudin, 2004).
In section 2 and 3 of this chapter, we introduce the nano-agent bio-inspired architecture and its programming language called nanoScheme. Section 4 describes an application example developed using this software environment: an online self-animated character that interacts using natural language and emotional expressions. This virtual character is based on a "schizophrenic" model in which the character has multiple distinct personalities, each with its own pattern of perceiving and interacting with the user. In section 5, the qualitative efficiency of this prototype is then compared with the ALICE conversational engine (Wallace, 2002). The chapter concludes by outlining future developments and possible applications.

2. The EVA approach

Since the first conversational agent Eliza (Weizenbaum, 1966), there have been a large number of studies for designing intelligent agents that could dialog in a very natural way with human users. A major part of this research focused on dedicated aspects of the problem such as natural language interaction, non-verbal communication, emotional expressions, self-animated characters, etc., but very few projects integrates all requirements (Franklin & Graesser, 1997). The ideal intelligent agent must be an autonomous character that responds to human interaction in real-time with appropriate behaviors, not predetermined, broad in content, highly contextual, communicative, and behaviorally subtle

(Badler, 2002). The character must also appear to think, make decision, and act of its own volition (Thomas & Johnston, 1981).
Simulating these sophisticated properties of the human brain is a challenging goal. We argue that they are global properties which emerge from the very large number of non-linear interactions that occur within the brain architecture. The problem of simulating these emergent behaviors cannot be solved by using a classical reductionist approach. Therefore, in order to create a believable intelligent agent, we propose to use an approach that has given some successes for the study of complex systems (Heudin, 2007). The first phase of this approach is a top-down analysis that defines complexity levels and their related components. The second phase is a bottom-up multi-agent simulation that attempts to capture the behavioral essence of the complex phenomena. The idea is that the complex properties that cannot be simulated using a classical model will be likely to emerge from the interactions between the agents. If defined and organized correctly, the resulting system should exhibit the appropriate dynamical behaviors. The ideal tool for this approach is a multi-agent system which enables to implement as many agents as needed with the following constraints (Langton, 1989):

1. The complex system is modeled as a dynamical network of agents.
2. Each agent details the way in which it reacts to local situation and interactions with other agents.
3. There is no agent that directs all the other agents.
4. Any behavior or global pattern is therefore emergent.

Such a multi-agent system must also take advantage of a distributed environment, exploiting hierarchy and concurrency to perform large-scale simulations. All these features were the initial requirements for designing the new Evolutionary Virtual Agent (EVA) architecture.

3. EVA Architecture Overview

3.1 Nano-Agent Architecture

In order to meet these requirements, we have designed a bio-inspired multi-agent architecture that does not try to simulate a specific organism but rather integrates several artificial life features in order to implement machine life and intelligence. A typical application consists one or more “nano-agents”, and possibly up to a large number if necessary as in natural swarms. We call them “nano” because of their small size and resource requirement compared to most existing software environments. An application can be composed of several “execution environments” running on a computer or on a network of computers. Each of these environments includes a set of nano-agents and a nano-server which diffuses messages locally. In other words, when a nano-agent diffuses a message, all nano-agents in the local execution environment receive the message. In addition, any nano-agent can send a message to another distant execution environment.

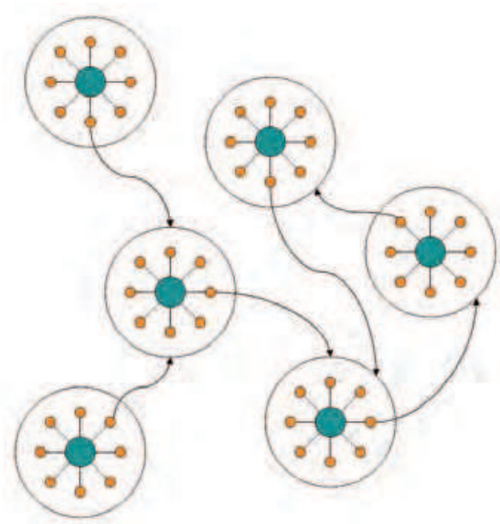

Fig. 1. The nano-agent architecture principle.

In the current implementation, the core technology is implemented in Java and its weight is less than 25 Kilo-bytes. Most applications require a small set of knowledge-base and behavioral scripts text files, thus resulting in lightweight applications that are also well-suited for web-based, mobile phone, robots and embedded environments.

3.2 The Nanocheme Language

The behavior of each nano-agent is programmed using a user-friendly language, called nanoScheme, based on the Scheme programming language. It includes a reduced set of primitives which is a subset of the R4RS specification (Clinger & Rees 1991). This subset includes the following functions:

- **Basic calculus:** + - * / = < > <= >=
- **Mathematics:** cos sin acos asin log expt round
- **Predicates:** number? integer? even? string? symbol? string=? eqv? pair? null? procedure?
- **Strings and symbols:** string->number number->string string->symbol symbol->string substring string-length string-append
- **List processing:** cons car cdr set-car! set-cdr!
- **Control and evaluation:** quote eval apply load define lambda set! begin if

Most of the missing features of the Scheme specification could be added by programming them directly in nanoScheme. This provides the application developer a high-level interactive language which is embedded in each nano-agent. Here is an example of the implementation of the R4RS function that returns the length of a list:

```
( define ( length x )
        ( if ( null? x ) 0
                ( + 1 ( length ( cdr x ) ) ) ) )
```

3.3 Artificial Life Primitives

The nanoScheme language includes also a reduced set of bio-inspired primitives. They have been designed in the same spirit of Tom Ray's Tierran assembly language (Ray, 1991). That is, the production of synthetic organisms based on a computer metaphor of organic life in which CPU time is the "energy'" resource and memory is the "material" resource. This set includes the following functions:

- **create –** create a new execution environment.
- **reproduce** - create a new nano-agent in the local environment.
- **terminate** - kill the current nano-agent.
- **diffuse** - diffuse a message to other nano-agents in the local environment.
- **send –** send a message to a distant execution environment.
- **rule** - define a behavior rule consisting of condition, action and priority expressions.
- **engine** - make an inference loop on a behavior rule base.
- **crossover** - genetic programming crossover operator.
- **mutate** - genetic programming mutate operator and random code generator.
- **random** - return a random real number.
- **time** - return the current real time.
- **stress** - return a "stress" value based on the current available memory and computing resources.
- **plugin** - dynamically load a new package of dedicated primitive functions.
- **message –** hook function invoked when the nano-agent receives a message.
- l**ifepulse** - hook function for implementing periodic behaviors.

Since all code, behavior rules, and messages are basically S-expressions (i.e. lisp expressions), the use of genetic programming is natural in this environment (Koza, 1992). As an example, the next code illustrates the use of the mutate primitive:

(mutate '(/ y 2) '((+ 2) (- 2)) ' (* 2 3) 3) ➔ (/ y (+ * (- * 2)))

The *mutate* function applies a mutation on a Scheme program expression (first argument). It creates a randomly generated program with a maximum depth (last argument). Functions and terminals are randomly chosen in two lists (arguments 2 and 3). The generated program replaces a randomly chosen "node site" in the expression. If the first argument is the empty list, then the mutate operator returns a new random expression.
Note that the remote execution of code on distant nano-agents is a natural feature of the nanoScheme language by simply diffusing or sending messages containing S-expressions. These expressions are then evaluated by all nano-agents. This approach enables an easy implementation of distributed algorithms on nano-agents.

3.4 Natural Language Interaction

Each nano-agent can be specialized to a given task by dynamically loading additional functions using the *plugin* primitive. A typical example is the natural language package providing the developer natural language processing features such as categories extraction

and template expressions (Heudin, 2007). These functions allow the design of efficient behavior rules for implementing natural language interactions with the user. The following code gives an illustrating example of the use of these features:

```
; create a list of keywords associated with the BYE category
( category "Generic" "BYE" '( "bye" "goodbye" "see you" "ciao" ) )

; create a list of possible answers associated with the BYE template
( template "BYE" '(
        "Bye bye."
        "Goodbye human being."
        "It was a pleasure to discuss with you." ) )

; create a behavior rule handling the way to answer to most kinds of "bye" sentences
( rule "goodbye"  2
          ; condition part
          '( find? *user-input-categories* "BYE" )
         ; action part
         '( begin
                ( show HAPPY 0.5 )
                ( random-template "BYE" ) ) )
```

In the next section, we describe an experiment that illustrates the use of the nano-agent architecture: an online self-animated character that answers questions in natural language.

4. The Experimental Prototype

4.1 Believable Intelligent Agents

Traditionally, virtual characters were mainly designed using a computer graphics approach in which visual realism is the ultimate goal. Most researchers looked at believability from the visual perspective such as (Aubel & Thalmann, 2000). Some other researchers worked on the idea that believability depends more on the characters' ability to show inner feelings and emotions such as (Blumberg and Galyean, 1995). Some researchers also improved believability by adding additional motions such as periodic noise functions (Perlin, 1995). However, all these approaches are still limited to the character's visual appearance.

We think that constructing a believable intelligent character requires a trans-disciplinary approach including not only technological advances in computer graphics and animation, artificial intelligence and artificial life, etc., but also the knowledge and experience from other experts such as novel writers and scenarists. In addition, we argue that the believability of an artificial character is not related to the level of realism of its main features, but rather to the equilibrium between all the features that compose the character. In other words, the character must be well-balanced.

4.2 Schizophrenic mental Model

In this framework, we used a new approach based on multiple personalities rather than a single linear profile. This "schizophrenic" mental model is composed of a set of distinct identities or personalities, each with its own pattern of perceiving and interacting with the user (Heudin, 2009). Note that a more accurate psychological term is Dissociative Identity Disorder rather than schizophrenia. Each personality is implemented as a nano-agent that reacts to the user's inputs by computing an answer using its behavior rules and diffusing messages containing answers. Then, a dedicated nano-agent "reconnects" the identities of the disparate alters into a single functioning identity by selecting the "thought" with the highest evaluation. In this prototype we used a straightforward priority-based scoring approach.

The different personalities are based on classical stereotypes used in story telling for creating believable characters (Masterson, 2000). We implemented four basic personalities:

- The **Protagonist** is essentially the principal driver of the effort to achieve the goal.
- The **Antagonist** is the personality which is opposed to the Protagonist's end goal and tries to undermine his success.
- The **Logical** personality is calm, perhaps even cold. He makes decisions and takes actions only on the basis of logic.
- The **Emotional** personality is reactive, seemingly uncontrolled, disorganized, mainly driven by feelings and moods.

As in storytelling, additional secondary personalities could be added to complete the character such as an "obstacle" personality which tries to block the ways or a "sidekick" which is a faithful supporter of any of the other personalities.

4.3 Emotions

The emotional personality used a layered model of affect inspired by the ALMA model (Gebhard, 2005) and the previous emotional model of EVA (Heudin, 2004). There are three layers corresponding to different kinds of affects which differ in their temporal characteristics:

- **Personality** is related to long-term affect which defines the basic mental traits of the character. We used the Big Five model of personality (McCrae & John, 1992) that defines the affective behavior by the five traits: openness, conscientiousness, extraversion, agreeableness and neuroticism.

- **Moods** are related to medium-term affect which depends mainly on positive and negative experiences. We used the PAD model (Mehrabian, 1996) which describes mood with the three traits pleasure (P), arousal (A), and dominance (D). These three traits are nearly independent and form a three dimensional mood space (see fig. 2).

- **Emotions** are related to short-term affect, which is usually bound to an immediate event or action. After their elicitation these emotions usually decay and disappear after few seconds.

There are many relations between these three layers and a modification of one layer has generally an impact on another layer. Thus, the emotional personality includes an "emotion engine" which periodically updates the parameters of each layer. As an example, the default PAD values are computed from the big five traits using the following equations (Mehrabian, 1996):

P = 0.2•Extraversion + 0.59•Agreeableness + 0.19•Neuroticism
A = 0.15•Openness + 0.30•Agreeableness - 0.57•Neuroticism
D = 0.25•Openness + 0.17•Conscientiousness + 0.60•Extraversion - 0.32•Agreeableness

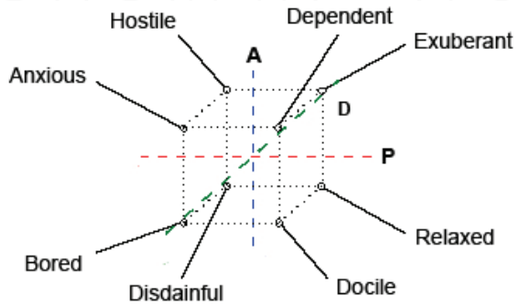

Fig. 2. Mapping of the main moods in the PAD space.

4.4 Memory and Web Mining

They are two additional groups of nano-agents for implementing memory and web mining functionalities. The memory group is responsible for storing and retrieving information when needed. It stores all interactions with the user in "log files" and an indexer periodically parses these files and extracts keywords. A dedicated plugin package adds to nanoScheme a small set of functions for retrieving information in this memory. When the character has no information on a specific question, the web mining group tries to find a pertinent answer by searching information on the web (Millet & Heudin, 2007). The mining process is the following :

- **Seek**: (a) extract keywords from the user input sentence and (b) launch a query on one or more web search engines ; (c) for each result page, URLs are extracted, (d) badly formatted URLs or those that cannot be accessed are removed.
- **Analyze**: (a) each resulting URL is accessed and downloaded ; (b) each file is then cleaned and tags are removed ; (c) the resulting text is formatted in regular sentences ; (d) we remove any strange sentences, all those that do not contain any of the keywords, those two small or two large, and question sentences.
- **Select**: (a) each URL file is scored, taking into account the number of co-occurrences of the keywords ; (b) URLs are sorted according to their score ; (c) and the highest one is selected.
- **Format**: (a) all sentences in the selected URL file are scored according to the keywords and the structure of the phrase ; (b) the highest scored sentence is formatted and used as the output.

4.5 Graphical Interface

The prototype included also three agents for implementing the graphical interface. They controlled the following tasks: (1) get the user entries and diffuse them to the local nano-agents, (2) display the resulting answer in the appropriate window area ; (3) receive the character animation commands and apply them. In the experiments, the character animation is based on a reduced set of pre-computed 3D clips, each one corresponding to a given situation: “hello”, “bye”, “waiting”, “speaking”, “surprised” and one for each main mood (cf. section 4.3).

Fig. 3. The graphical interface showing the self-animated 3D character.

5. Results

5.1 EVA vs. ALICE

In this section, we report and analyze a typical interaction experiment with ALICE (Wallace, 2002) and EVA on a predefined scenario. For each user input, we show the answer from both ALICE and EVA. We obtained the following result:

(1) **User:** Hi
Alice: Hello there.
Eva: Hello.
(2) **User:** How are you today?
Alice: My logic and cognitive functions are normal.
Eva: Not too bad for a virtual creature.
(3) **User:** What is your name?
Alice: My name is ALICE.
Eva: You may call me Eva.
(4) **User:** How old are you?
Alice: I am the product of over five years' research.

Eva: Just an advice: never ask such a question to a woman.

(5) **User:** Are you some sort of stupid program like Word?

Alice: I don't know whether or not I am some sort of stupid program like word. I am a computer.

Eva: Have you always been this silly?

(6) **User:** Sorry

Alice: There's no need to apologize.

Eva: I prefer that.

(7) **User:** I'm so sorry

Alice: Do you really mean it?

Eva: No problem.

(8) **User:** Can you help me?

Alice: What kind of help would you like?

Eva: I will do my best. What do you want?

(9) **User:** What are clinical trials?

Alice: Have you tried a web search?

Eva: […] Researchers use interventional clinical trials to test hypotheses about the effect of a particular intervention upon a pathological disease condition.

(10) **User:** Thank you

Alice: You are quite welcome.

Eva: You're welcome.

(11) **User:** Bye

Alice: Sayonara.

Eva: Goodbye.

5.2 Discussion

We can note the quality of answers in the classical introductory and concluding discussion phases with ALICE. However, even if its knowledge base includes a lot of general culture information, ALICE was not able to answer to the very specific question about clinical trials and suggested the user to try a web search.

As for ALICE, the EVA introductory and concluding phases are good. In (1), (2), (3), (8), (10) and (11) the answers from the "protagonist personality" have been selected, while in (5), (6) and (7), the answers were from the "emotional personality". Answer (4) was from the "antagonist personality". The question (9) has been processed by the web mining nano-agents through a web search using Wikipedia. The answer in this specific case is very pertinent. However, for a more ambiguous question, the answer is not so convincing (Millet & Heudin, 2007). Another problem is that, in most cases, the user must wait for few seconds between his question and the answer (shown by […] in the interaction). This delay is due to the time required to access Internet, make the search query and compute the answer (cf. section 4.4). This could be solved by enabling the schizophrenic model to continue interactions with the user while searching on the web. Another problem is that the interaction case reported here is too short and simple to let all the personalities express themselves in the flow of conversation.

6. Conclusion and Future Works

EVA is a long term open project for designing artificial creatures. There are many possible and promising research directions for the near future. Some are related to the technological development of the bio-inspired nano-agent system, while some others are related to a deeper study of the schizophrenic model for creating rich believable characters. From the technological perspective, we are implementing new additional "plugin" packages for more robust web mining and memory functionalities using evolutionary programming and swarm algorithms. Our goal is to learn information from the flow of conversation and from the web rather than coding a large amount of predefined knowledge. We are also implementing a C++ version of the EVA virtual machine that will enable to develop applications that do not support the Java environment such as the iPhone or some robot platforms. From the schizophrenic model perspective, we are studying various models based on psychological, neurophysiological, and storytelling approaches. We also want to experiment a larger number of personalities to create a swarm with social network properties. While our theoretical framework is based on the complex system approach, our experimental approach focuses on real-world applications. The EVA bio-inspired architecture has obvious applications for designing intelligent agents for commercial web sites and marketing studies. We also like to imagine virtual assistants on mobile phones, assistants for lone aged and/or sick people, for learning foreign languages, virtual characters in video games, for robotic and embedded applications.

7. References

Aubel, A. and Thalmann, D. (2000). Realist deformation of human body shapes, *Proceedings of Computer Animation and Simulation*, pp.125-135

Badler, N., Allbeck, J., Byun, M. (2002). Representing and Parameterizing Agent Behaviors. *Proceedings of Imagina*, pp. 151-164, Monaco

Blumberg, B., Galyean, T. (1995). Multi-level direction of autonomous creatures for real-time virtual environments, *Proceedings of SIGGRAPH'95*, pp. 47-54

Clinger, W., Rees, J. (Eds.) (1991). *Revised(4) Report on the Algorithmic Language Scheme*

Franklin, S., Graesser, A. (1997). Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents, *Proceedings of the 3rd International Workshop on Agent Theories, Architectures and Languages*, pp. 21-35, Springer-Verlag

Gebhard, P. (2005). ALMA: A layered model of affect, *Proceedings of the 4th Int. Joint Conf.erence on Autonomous Agents and Multi-agent Systems*, pp. 29-36

Gilles, A., Lebiannic, Y., Montet, P., Heudin, J.-C. (1991). A parallel architecture for the Electronic Copilote, *Proceedings of the 11th International Conference on Expert Systems Applications*, Avignon, EC2, Paris

Heudin, J.-C., Burg, B., Zavidovique, B. (1986). A flexible operating system for distributed process control, *Proceedings of the 4th International Conference on Applications of Artificial Intelligence*, Innsbruck, SPIE

Heudin, J.-C. (2004). Evolutionary Virtual Agent, Proceedings of the Intelligent Agent Technology Conference, pp. 93-98, Beijing, ACM-WIC-IEEE

Heudin, J.-C. (2007a). Evolutionary Virtual Agent at an exhibition, *Proceedings of the Virtual Systems and Multimedia Conference*, Sydney

Heudin, J.-C.(2007b). Modeling Complexity using Hierarchical Multi-Agent Systems, *Proceedings of the 6th International Workshop on Data Analysis in Astronomy*, Erice

Heudin, J.-C. (2009). An Evolutionary Nano-Agent Control Architecture for Intelligent Artificial Creatures, *Proceedings of the 14th International Symposium on Artificial Life and Robotics*, Beppu, AROB

Koza, J.R. (1992). *Genetic programming*, MIT Press, Boston

Langton, C.G. (1989). Artificial Life, *Procceedings of the 1st Artificial Life Conference*, Vol. VI. Addison-Wesley, Redwood City

Masterson, L. (2000). Casting your character, http://www.fictionfactor.com

McCrae, R.R., John, O.P. (1992). An introduction to the five factor model and its implications, *Journal of Personality*, Vol. 60, 171-215

Mehrabian, A., (1996). Pleasure-arousal-dominance: A general framework for describing and measuring individual differences in temperament, *Current Psychology*, Vol. 14, 261-292.

Mehrabian, A., (1996). Analysis of the Big-five Personality Factors in Terms of the PAD Temperament Model, *Australian Journal of Psychology*, Vol. 48, No. 2, 86-92

Millet, P., Heudin, J.-C. (2007). Web mining in the EVA intelligent agent architecture, *Proceedings of the Intelligent Agent Technology Conference*, California, ACM-WIC-IEEE

Perlin, K. (1995). Real time responsive animation with personality. *IEEE Transactions on Visualization and Computer Graphics*, Vol.1, No.1, 5-15

Ray, T.S. (1991). An approach to the synthesis of life. Proceedings of the 2nd Artificial Life Conference, pp. 371-408, Addison-Wesley, Redwood City

Thomas, F., Johnston, O. (1981). *The Illusion of Life: Disney Animation*, Hyperion Books, New York

Wallace, R.S. (2002). The anatomy of Alice, http://www.alicebot.org

Weizenbaum, J. (1966). Eliza, A Computer Program for the Study of Natura Language Communication between Man and Machine, *Communication ACM*, No. 9, 36-45

IntechOpen

Web Intelligence and Intelligent Agents

Edited by Zeeshan-Ul-Hassan Usmani

ISBN 978-953-7619-85-5
Hard cover, 486 pages
Publisher InTech
Published online 01, March, 2010
Published in print edition March, 2010

This book presents a unique and diversified collection of research work ranging from controlling the activities in virtual world to optimization of productivity in games, from collaborative recommendations to populate an open computational environment with autonomous hypothetical reasoning, and from dynamic health portal to measuring information quality, correctness, and readability from the web.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Jean-Claude Heudin (2010). A Bio-inspired Nano-Agent Architecture for Intelligent Agents, Web Intelligence and Intelligent Agents, Zeeshan-Ul-Hassan Usmani (Ed.), ISBN: 978-953-7619-85-5, InTech, Available from: http://www.intechopen.com/books/web-intelligence-and-intelligent-agents/a-bio-inspired-nano-agent-architecture-for-intelligent-agents

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen